Concurrent Application Testing Based on a Throughput Metric

John T. Daly, HPC-4

The Advanced Simulation and Computing (ASC) program has produced various metrics-based approaches for evaluating the effectiveness of a high-performance computing (HPC) system, however these typically reflect the system perspective [1]. What is lacking is a consistent set of accessible and intuitive metrics that accurately capture the application's perspective. By articulating the application throughput in terms of node performance, parallel scaling, checkpoint efficiency, and operational utilization on specified numbers of allocated nodes as follows, one can formulate requirements verification in terms of a new paradigm of concurrent application testing [2].

Application Throughput = (Performance • Scaling • Efficiency • Availability) $_{\textbf{Allocated Nodes}}$

Under this new paradigm, the performance and scaling data collected during system integration would be combined with efficiency and utilization data collected on the production platform to validate a system against specific application throughput requirements [2]. In this way, the bulk of the testing can be performed in concurrence with actual production workload.

By so doing, one increases the productive lifetime of the system by shifting weeks or even months of testing out of the integration phase. Typically, acceptance testing for a new system involves running hundreds of hours of simulated problems. By performing concurrent application testing, the bulk of this data would be based on actual production jobs. This has the added advantage of increasing the certainty of the test results by basing them on data gathered over longer run times generated by a real workload. In particular, the efficiency and utilization are substantially governed by the application and system mean time to interrupt (MTTI) and mean time to repair (MTTR) [3]. Because these values are based on system events that occur randomly [4], measuring them over weeks or even months gives a better approximation of the expected values.

Figure 1 illustrates the concept of concurrent application testing using actual Red Storm (ASC supercomputer at Sandia National Laboratories) checkpoint efficiency results generated over a 5-month period from the run of a production problem on that system. The results demonstrate both the improvement of checkpoint efficiency over time, and the high level of agreement obtained between the model and actual data measured over significant time periods.

For more information contact John Daly at jtd@lanl.gov.

[1] J.R. Stearley, "Defining and Measuring

Supercomputer Reliability, Availability, and Serviceability (RAS)," http://www.cs.sandia.gov/~jrstear/ras (2005).
[2] J.T. Daly, "Methodology and Metrics for Quantifying Application Throughput," Proceedings of the Nuclear Explosives Code Developers' Conference (NECDC), in press.
[3] J.T. Daly, "Evaluating the Performance of a Checkpointing Application Given the Number and Types of Interrupts," Proceedings of the First Workshop on High Performance Computing Reliability Issues (HPCA-11), San Francisco, (2005).
4] J.T. Daly, "Failure Analysis From the Application's Point of View", Los Alamos Technical Report, LA-UR-06-8089.

Funding Acknowledgements
This research was supported by the NNSA tri-Lab Advanced Simulation and Computing Program.

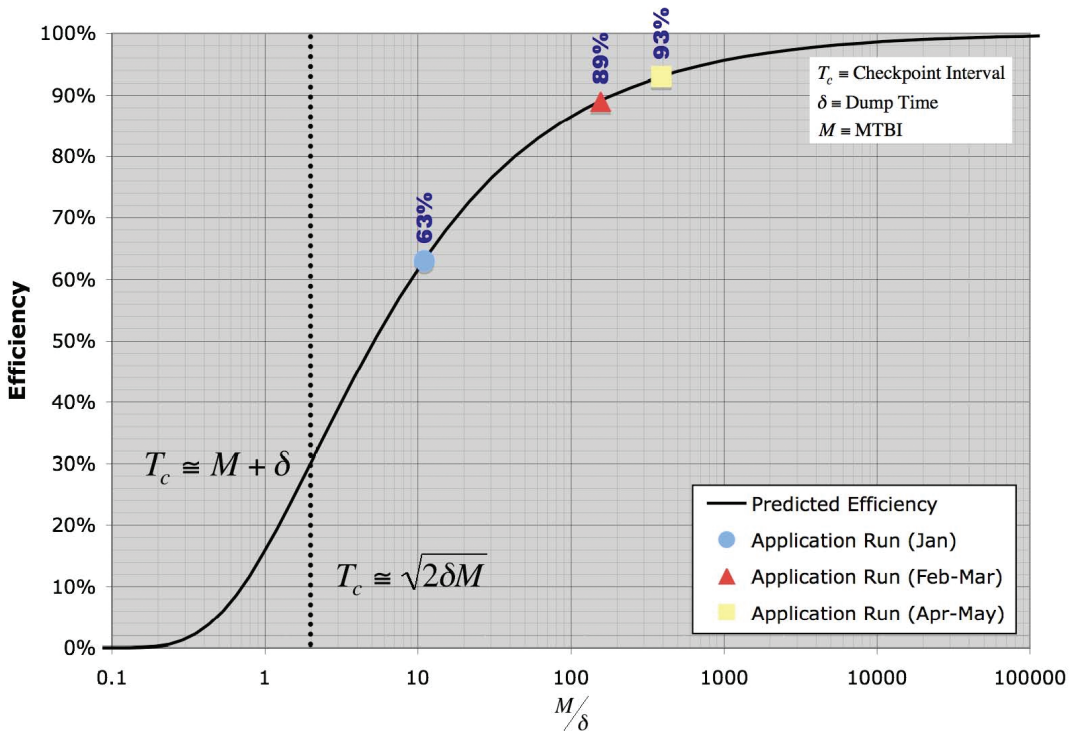

Fig. 1.
Checkpoint efficiency as a function of dump time, application MTTI, and the restart overhead in blue. The optimum checkpoint interval is plotted in red to demonstrate its asymptotic behavior for large and small values of the nondimensional dump time.